Superpixel Segmentation With Edge Guided Local-Global Attention Network

Mingzhu Xu, *Member, IEEE,* Zhengyu Sun, Yijun Hu, Haoyu Tang, *Member, IEEE,* Yupeng Hu, *Member, IEEE,* Xuemeng Song, *Senior Member, IEEE,* and Liqiang Nie, *Senior Member, IEEE*

***Abstract*—Superpixel segmentation aims to automatically group visually similar pixels within an image into compact regions. This approach provides an efficient low-level representation of image data, effectively reducing the complexity of image primitives for subsequent vision tasks. Recent deep convolutional networks have shown their advantages in superpixel segmentation task. However, many existing deep learning methods still struggle to preserve object edges and accurately perceive similar pixels. This limitation can be attributed to their inadequate ability to model edge information and capture effective context within the image. To address these issues, we propose an Edge guided Local-Global Attention Network (ELGANet) for superpixel segmentation. Specifically, we first devise an Edge Enhancement Module (EeEM), which integrates multiple edge features into the superpixel-friendly features. Then, we develop a Local-Global Attention Module (LGAM) to analyze the relationship between pixels and local or global region patches, expecting to obtain effective context information for grouping similar pixels. The edge features and deep global semantic features are subsequently fused to generate the superpixel-friendly features. The final superpixel-friendly features are then mapped into final superpixels. Extensive experiments on four benchmark datasets demonstrate the effectiveness and superiority of our ELGANet compared with ten state-of-the-art models.**

***Index Terms*—Superpixel segmentation, Edge enhancement, Local-Global context**

I. Introduction

SUPERPIXEL segmentation aims to over-segment an image into a series of compact regions, which is generated by clustering perceptually similar pixels based on low-level image properties [1], [2]. Different from the isolated pixels in digital images, which are numerous and lack semantic information, superpixels offer a more semantically meaningful and efficient representation of image data. They provide more semantically informative processing primitives for subsequent visual tasks, and greatly improve the efficiency of computer vision tasks, such as salient object detection [3]–[7], object tracking [8], [9], and semantic segmentation [10]–[12].

A common practice for superpixel segmentation is to first divide the image into a series of patches as initial superpixels. Then, perform iterative analysis of the similarity of features between pixels and surrounding superpixels, estimating the affiliation of each pixel to its surrounding superpixels until stability is reached [2]. Traditional methods rely on manually crafted features to group similar pixels. The SLIC-like methods [13], [14] employ common iterative clustering approach, while graph-based methods [15]–[19] treat segmentation as a discrete optimization problem. While traditional superpixel methods have made significant advancements, they are typically non-differentiable, posing challenges for their integration into deep learning models for end-to-end training of established visual tasks. This limitation restricts their applicability.

Recently, some deep learning-based superpixel models [20]–[22] have been proposed, leading to a significant improvement in superpixel segmentation performance. In these deep superpixel methods, the extraction of deep features tailored for superpixels and the establishment of an association score matrix between pixels and neighboring superpixels are pivotal for effective superpixel segmentation. For the superpixel-friendly deep features, most methods employ the encoder-decoder architecture to extract multi-scale contextual information [20], [21]. Additionally, the Association Implantation Module (AIM) [22] has been developed to enrich pixel features by consolidating information from neighboring grid cells through convolution operations. For the association score matrix, Superpixel Sampling Networks (SSN) [20] transforms the non-differentiable iterative association mapping into a differentiable one by transitioning pixel hard assignments to pixel soft assignments. This enables the model to adapt its superpixel representations according to the specific task requirements, albeit at a increased computational complexity due to iterative processes. The Full Convolution Network (FCN) [21] directly learns the pixel-superpixels association maps through convolution layers with supervision from datasets and loss functions. While this approach achieves faster computation speed, it may struggle to learn diverse superpixel representations for different vision tasks. Although these existing deep superpixel networks have achieved good performance, they still encounter challenges in accurately preserving object edges and discerning similar pixels.

To tackle these challenges, we propose an Edge guided Local-Global Attention Network (ELGANet), comprising two purposefully crafted modules: the Edge Enhancement Module (EeEM) and the Local-Global Attention Module (LGAM). The EeEM module is designed to explicitly model edge infor-

(Corresponding author: Yupeng Hu, Xuemeng Song.)

Mingzhu Xu, Zhengyu Sun, Yijun Hu, Haoyu Tang, and Yupeng Hu are with the School of Software, Shandong University, JiNan 250101, Shandong, China (e-mail: xumingzhu@sdu.edu.cn, sunzy53@gmail.com, yijunhu60@gmail.com, tanghao258@sdu.edu.cn, huyupeng@sdu.edu.cn).

Xuemeng Song is with the School of Computer Science and Technology, Shandong University, Qingdao, Shandong 266237, China (e-mail: sxmustc@gmail.com).

Liqiang Nie is with the School of Computer Science and Technology, Harbin Institute of Technology (Shenzhen), ShenZhen 518055, Guangdong, China (e-mail: nieliqiang@gmail.com).

mation, aiming to enrich feature representation and enhance the edge accuracy of deep superpixel networks. Besides, the LGAM module aims to augment pixel contextual information by analyzing the interplay between pixels and local neighboring patches, as well as global patches. This enhancement boosts the network's capability to perceive similar pixels and promotes improving superpixel compactness. Specifically, as illustrated in Fig.1, the superpixel-friendly features are extracted from five encoder stages. The edge enhanced feature are generated in stage 2, where multiple edge features are integrated into the superpixel-friendly features. Then, from stage 3 to stage 5, we extract the local and global context information by analyzing the relationship between pixels and local surrounding patches, as well as global patches. Finally, the edge features and deep global semantic features are fused to refine the superpixel-friendly features at various resolutions from deep to shallow layers. The final superpixel-friendly features generated from decoders are mapped into superpixels through one mapping operation. Extensive experiments on four datasets demonstrate the effectiveness and superiority of our ELGANet compared with ten state-of-the-art models.

We summarize the main contributions as follows:

1. We propose a novel Edge Enhancement Module (EeEM) that explicitly models multiple edge information and integrates them into superpixel feature representation. This module refines superpixel features at various resolutions, thereby enhancing the edge accuracy of deep superpixel networks.

2. We propose a novel Local-Global Attention Module (LGAM) that explores the interaction between pixels and their local surrounding patches, as well as global patches. This module enhances the network's ability to perceive similar pixels and promotes improving superpixel compactness.

3.We also conduct comprehensive experiments on four benchmark datasets. Through extensive experimental evaluations and ablation studies, we validate the effectiveness of our key modules and demonstrate the superiority of our ELGANet when compared to ten state-of-the-art models. The source codes and trained models will be released upon acceptance.

II. Related Works

A. Traditional superpixel methods

Traditional superpixel methods group similar pixels based on low-level features. These methods are typically categorized into graph-based methods and clustering-based methods. Graph-based methods formulate superpixel segmentation as a graph partitioning problem, with pixels serving as vertices and edges indicating the similarity between connected pixels. Superpixels are formed as a series of subgraphs through solving a discrete optimization problem. Pixels sharing similar features belong to the same subgraph, while those with different features belong to different subgraphs. FH [15], RW [16], ERS [17], and the recent DRW [18], HSSPCL [23] are the representative graph-based methods. Different from the graph-based methods, the clustering-based methods treat superpixel segmentation as a pixel clustering problem. Initially, the image is divided into a series of patches to serve as initial superpixels. Then, the algorithms iteratively estimate each pixel's affiliation to its surrounding superpixels until stability is achieved. Classic methods like Simple Linear Iteratively Clustering (SLIC) [2] restricts the search range of surrounding superpixels to spatially nearest neighbors. Other techniques, such as LSC [13] and Manifold-SLIC [14], explore more effective feature spaces for pixel clustering. FLIC [24] considers the neighboring continuity and explores an active search scheme rather a fixed search regions in SLIC. SNIC [25] proposes a non-iterative approach to growing superpixel clusters. SCSC [26] formulates the superpixels segmentation as a subspace clustering problem. ESOM [27] enhances superpixel segmentation by incorporating edge information. It employs a directional statistical ratio-based edge detector with Gaussian-shaped windows, and prevents edge pixels from merging into superpixels by evaluating pixel intensity dissimilarity, spatial distance, and edge cues, thereby improving boundary adherence. LAD [28] leverages image local standard deviation to enhance responses in low-contrast regions. VSSS [29] introduces a new pixel assignment scheme inspired by vine spread processes. Although traditional superpixel methods have made significant strides, their algorithms are typically non-differentiable. This presents challenges for integrating them into deep learning models for end-to-end training of established visual tasks, limiting their applicability.

B. Deep superpixel methods

Deep superpixel methods group similar pixels based on deep features, significantly enhancing superpixel performance. SEAL-ERS [30] introduces the first deep superpixel network by integrating a trainable pixel affinity network with existing graph-based superpixel segmentation techniques. However, the non-differentiable nature of the graph-based pixel-superpixel association prevents it from becoming a fully end-to-end learning network. SSN [20] develops a differentiable SLIC method, replacing pixel-superpixel hard assignments with a soft assignment scheme, making it the first end-to-end trainable superpixel network. FCN [21] directly learns pixel-superpixel association maps through convolution layers with supervision from datasets and loss functions. Another crucial issue is the extraction of superpixel-friendly features. AINet [22] introduces an Association Implantation Module (AIM) to enrich pixel features by consolidating information from neighboring grid cells. NLM [31] utilizes multiple parallel dilated convolutions to generate multi-scale cluster-friendly features. Additionally, for precise edge accuracy, ESNet [32] incorporates an additional edge loss to supervise latent superpixel features. Moreover, FSNet [33] integrates frequency domain sharp boundary information with multi-scale deep features to enhance superpixel performance. Despite the progress, existing deep superpixel networks still face challenges in accurately preserving object edges and distinguishing similar pixels.

C. Subsequent visual tasks

Superpixel are widely used in various visual tasks due to the ability to provide rich semantic and object structure information. For salient object detection, the authors in [3]

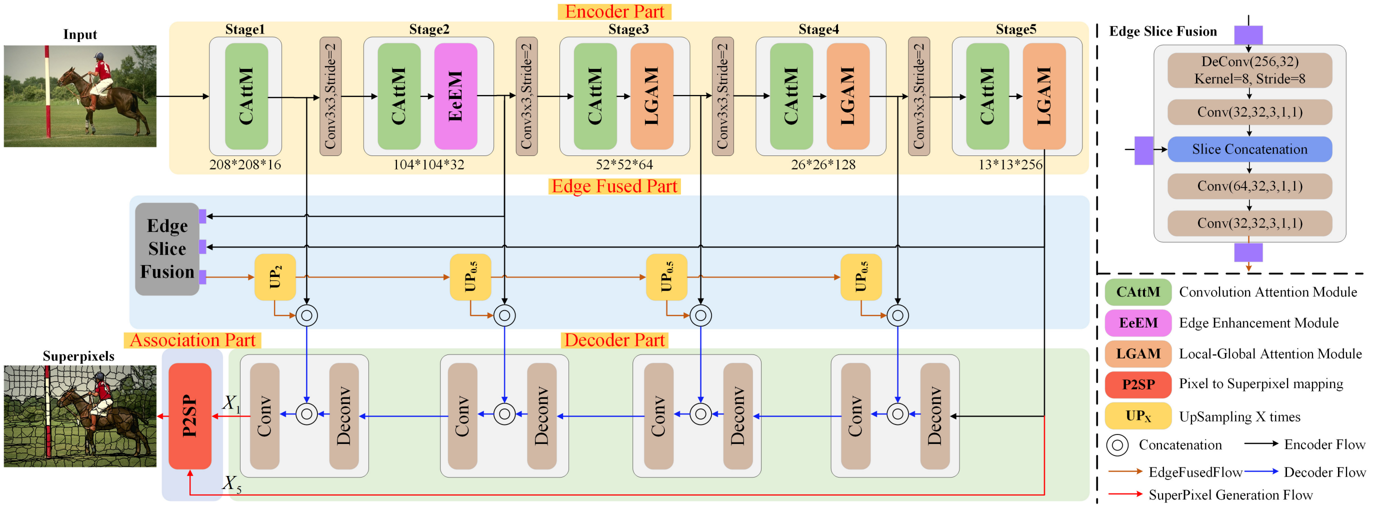


Fig. 1: Our ELGANet consists of four main parts. The Encoder Part is dedicated to extract superpixel-friendly features with the assistant of our EeEM and LGAM modules. The Edge Fused Part generates edge-enhanced local-global contextual features. The Decoder Part progressively refines features from deep to shallow layers to generate the final superpixel-friendly features. The Association Part generates the final superpixels.

established a graph using superpixel and apply graph clustering to address the video salient object detection task. Cai et al. [4] leveraged the sensitivity of superpixel to low-level information to compensate for the information lost in high-level feature, thereby enhancing the accuracy of target boundaries. Superpixel is utilized as graph nodes in [5] to establish a video salient object detection framework based on graph convolutional networks. In [6], authors proposed a multi-level superpixel-based target detection framework to enhance ship target detection. CCTNet [7] utilizes patch and superpixel tokens collaboratively to enhance the model's understanding of image semantics and object structure. In the field of object tracking, superpixel-based discriminative appearance model [8] is designed to enhance the model's ability to preserve the boundary information of both the target and the background. Zhan et al. [9] developed object tracking method based on salient superpixel, which are computed by manifold ranking. For the task of semantic segmentation, a novel framework presented in [10] employs superpixels to generate candidate pseudo-labels, which are then transformed using the invariant representations of shape and intensity derived from medical images. Jing et al. [34] proposed a fast SAR image segmentation method using superpixels. It first over-segments the image into superpixels, then automatically determines the cluster number via the density peak algorithm, and finally applies modified k-means clustering for efficient segmentation. Additionally, [11] introduces the utilization of superpixels to augment the creation of pseudo-labels for the weakly-supervised semantic segmentation.

III. Proposed Method

In this section, we present our Edge Guided Local-Global Attention Network (ELGANet) for superpixel segmentation. We begin by outlining the overall architecture in section III-A. Following that, in section III-B, we introduce our new Edge-Enhancement Module (EeEM). In section III-C, we delve into our proposed local-global attention module (LGAM). In section III-D, we describe the superpixel generation method. Finally, the training loss is provided in section III-E.

A. *Overall Architecture*

As depicted in Fig.1, our ELGANet comprises four main parts. The Encoder Part is responsible for extracting superpixel-friendly features, while the Edge Fused Part generates edge-enhanced local-global contextual features. The Decoder Part progressively fuses features from deep layers to shallow ones to generate the final superpixel-friendly features, and the Association Part generates the final superpixels. Specifically, given a raw input image, the superpixel-friendly features are extracted from five encoder stages of our **Encoder Part**. The edge-enhanced features are generated in stage 2, where multiple edge features are fused to generate edge enhanced features. Then, from stage 3 to stage 5, we progressively extract the local and global context information by analyzing the relationship between pixels and local surrounding patches, as well as global patches. To enhance the feature representations, the edge features and deep global semantic features are fused in the EdgeSliceFusion module (Details are illustrated in the upper right of Fig. 1), then the superpixel-friendly features are refined at various resolutions from deep to shallow layers in the **Edge Fused Part**. The **Decoder Part** utilizes simple deconvolution and convolution operations to generate the final superpixel-friendly features. In the **Association Part**, the final superpixels are generated through a single association mapping operation. Next, we first briefly introduce the Convolution Attention Module (CAttM) and the Slice Concatenation. Then, the details of our new Edge-Enhancement Module (EeEM) and Local-Global Attention Module (LGAM) will be described.

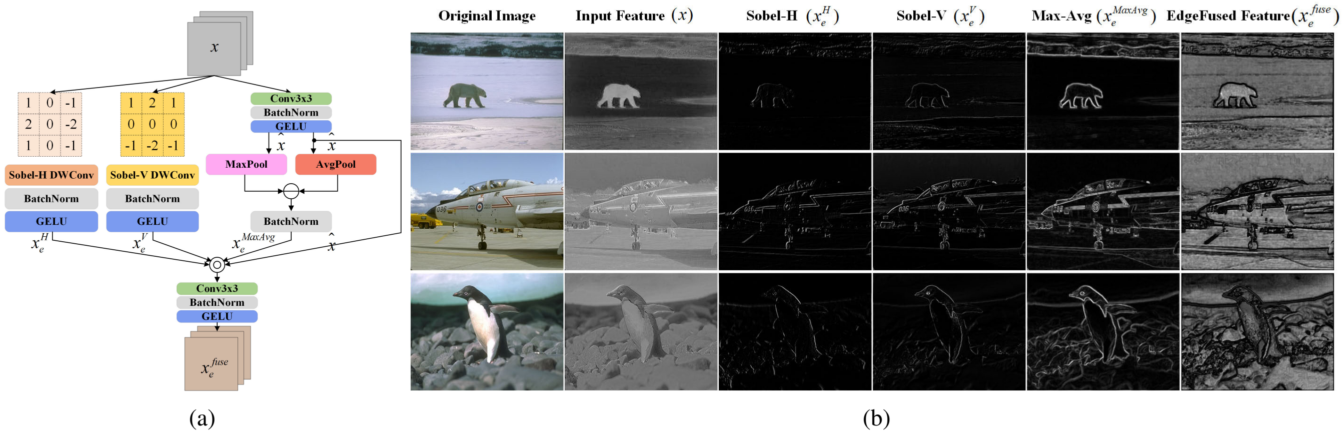


Fig. 2: Edge Enhancement Module (EeEM). (a) is the integral structure of our EeEM. (b) shows the visual examples of feature maps generated by different submodules.

The Convolution Attention Module (CAttM). Different from the plain basic convolution block, we adopt the Convolution Attention Module (CAttM) as the basic feature extractor, aiming to introduce inductive bias, cross-channel interaction, and lightweight design. The CAttM is inspired by architectures seen in SENet (Squeeze-and-Excitation Network) [35] and EfficientNetV2 [36], as outlined below,

$$x = BN(Conv_{1\times1}(SE(GELU(DWConv_{3\times3}(x))))) \tag{1}$$

where the $DWConv_{3\times3}$, $GELU$, SE, $Conv_{1\times1}$, and BN represent the 3×3 depth-wise convolution, Gaussian Error Linear Unit, Squeeze and Excitation block, 1×1 plain convolution, and Batch Normalization operations, respectively.

Slice Concatenation. In the Edge Fusion Part, the Slice Concatenation of the Edge Slice Fusion fuses the edge features and deep features. Unlike the standard concatenation operation which connects features group by group, Slice Concatenation merges these two types of features channel by channel in order to better integrate features from different stages. In practice, two different feature maps can be represented as $\{f_a^1, f_a^2, f_a^3, \ldots, f_a^n\}$ and $\{f_b^1, f_b^2, f_b^3, \ldots, f_b^n\}$. After performing the Slice Concatenation, the concatenated feature maps are $\{f_a^1, f_b^1, f_a^2, f_b^2, f_a^3, f_b^3, \ldots, f_a^n, f_b^n\}$.

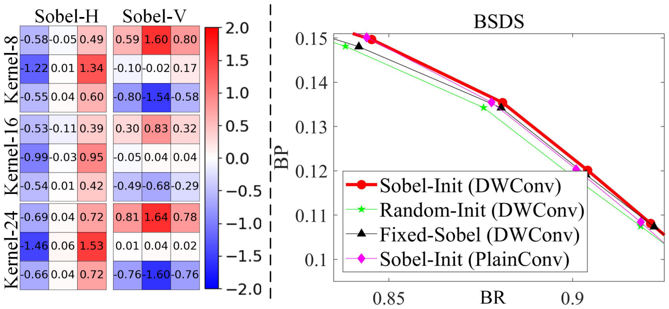


Fig. 3: (a) Visualization of the (8th, 16th, 24th) learned Sobel-H and Sobel-V kernels. (b) BP-BR curves of different Sobel configurations on the BSDS dataset.

B. Edge Enhancement Module (EeEM)

Edge information plays a crucial role in superpixel segmentation, helping to better adhere to boundaries. Existing ESOM [27], [34] method also leverages edge information to enhance segmentation. It employs traditional directional statistical ratio-based edge detectors and prevents edge pixels from merging into superpixels by evaluating pixel dissimilarity, effectively improving segmentation. Unlike this traditional approach, to develop a learnable method, we propose integrating learnable 'Sobel' operators, 'Max-Avg' change detection, and convolutional parameters to generate edges. It can be seamlessly incorporated into deep networks, enabling end-to-end learning, and is capable of extracting richer, more flexible edge information. Specifically, we propose a new Edge-Enhancement Module (EeEM), which is integrated into Stage 2 of the encoding part. This decision is based on the fact that shallow layers can capture finer edge details, and extracting features at a smaller image resolution (obtained by downsampling image features in Stage 1 by a factor of 2) reduces computational complexity. As illustrated in Fig. 2(a), three distinct branches are specifically designed to extract multiple types of edge features. The 'Sobel-H DWConv' branch is tailored to capture horizontal gradient information, while the 'Sobel-V DWConv' branch focuses on vertical gradient information. The 'Max-Avg' branch is dedicated to perceiving abrupt changes information. Subsequently, the outputs from these branches are concatenated with the original features, followed by a convolution block to enhance the edge guided superpixel features.

The 'Sobel-H DWConv' and 'Sobel-V DWConv' branches. As shown in Fig. 2(a), to capture the horizontal and vertical gradient information, we construct two learnable convolutional edge extraction branches: the 'Sobel-H DWConv' branch and 'Sobel-V DWConv' branch. Unlike traditional convolution, where learnable parameters are randomly initialized, we initialize the 'Sobel-H DWConv' and 'Sobel-V DWConv' with the horizontal and vertical convolution parameters of sobel edge detection. In addition, the depth-wise (DW) conv

This article has been accepted for publication in IEEE Transactions on Circuits and Systems for Video Technology. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TCSVT.2025.3587485

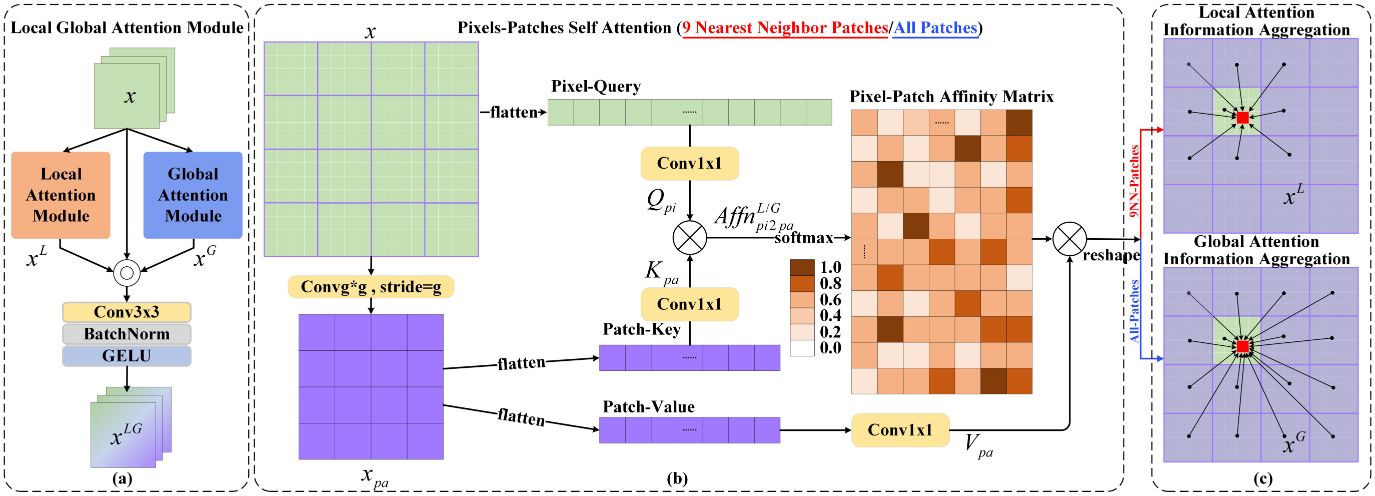


Fig. 4: Local Global Attention Module (LGAM). (a) is the internal structure of our LGAM. (b) illustrates the key pixels-patches self attention mechanism. (c) shows the local context and global context.

is adopted to decouple spatial and channel information. The sparsified filter connections, a form of regularization, promote focus on horizontal/vertical edges. These are then passed through batch normalization and activation functions to learn edge features, which can be summarized as follows,

$$\begin{aligned} x_e^{H/V} &= Sobel^{H/V} DWConv_{3\times 3}(x) \\ x_e^{H/V} &= GELU(BN(x_e^{H/V})) \end{aligned} \tag{2}$$

where $Sobel^{H/V} DWConv_{3\times 3}$ denotes horizontal sobel convolution or vertical sobel convolution with kernel size of 3×3. The operations BN and $GELU$ represent batch normalization and the activation function, respectively. The Sobel filters are learnable, starting from predefined values, to handle complex edges. The visualization of learned DW Sobel filters are shown in Fig.3(a), and BP-BR curves in Fig.3(b) show that our learnable DW Sobel filter outperforms that with non-learnable fixed parameter, randomly initialized, or plain conv sobel. All these observations validate the effectiveness of our proposed learnable sobel filters on extracting edge information.

The 'Max-Avg' branches. To diversify edge information, we specifically devise the 'Max-Avg' branch to detect abrupt changes within local windows. Initially, we pass the original input features through a standard convolutional block, which is formulated as follows.

$$\widehat{x} = GELU(BN(Conv_{3\times 3}(x))) \tag{3}$$

Subsequently, we perform max-pooling and average-pooling operations separately on these features, both with a kernel size of 3×3 and stride 1. Then, we compute the difference between the results of max-pooling and average-pooling, followed by batch normalization. Operations can be formulated as follows,

$$x_e^{MaxAvg} = BN(Maxpool_{3\times 3}(\widehat{x}) - Avgpool_{3\times 3}(\widehat{x})) \tag{4}$$

where $Maxpool_{3\times 3}$ and $Avgpool_{3\times 3}$ denote the max-pooling and average-pooling operations with kernel size of 3×3 and stride 1, respectively. This operation assigns high values to the local windows with abrupt changes in pixel values and low values to those without such changes. Consequently, this branch effectively detects abrupt changes within local windows and enriches the edge information.

Finally, the edge features from above branches and the basic features are fused to achieve the edge-enhanced feature representation x_e^{fuse}. The fusion computation can be formulated as follows,

$$x_e^{fuse} = GELU(BN(Conv_{3\times 3}(cat(x_e^H, x_e^V, x_e^{MaxAvg}, \widehat{x})))) \tag{5}$$

where $cat(\cdot)$ refers to the concatenation operation in channel dimension. Fig. 2(b) exhibits the visualized feature maps from different branches, which also validate the effectiveness of our EeEM in extracting edge sensitive features.

C. *Local Global Attention Module (LGAM)*

To enhance the network's ability to perceive similar pixels and improve the superpixels compactness, we propose a novel Local-Global Attention Module (LGAM) that thoroughly explores local and global information. As illustrated in Fig. 4(a), this module separately explores and fuses local and global context to obtain a superpixel-friendly feature representation. Superpixel segmentation determines pixel's affiliation by analyzing the correlation between pixels and surrounding patches. Inspired by this, we propose to generate superpixel-friendly pixel features by aggregating information from surrounding patches, as well as global patches. This approach differs from traditional Transformers [37], which typically aggregate information by analyzing correlations between pixels or between patches. Next, we first elaborate our key Pixels-Patches Self Attention mechanism, then briefly introduce the implementation of our local and global self attention.

Pixels-Patches Self Attention. We propose our new Pixels-Patches Self Attention (PiPaSA). The central pixel features are updated by aggregating information from local surrounding patches, as well as global all patches. This pixels-patches level context is more conducive to superpixel segmentation.

As shown in Fig. 4(b), given the input feature x, the pixel-level query vector Q_{pi} is first generated by flatten and linear transformation operation, which is formulated as follows,

$$Q_{pi} = Conv_{1\times1}(flatten(x)) \tag{6}$$

where $flatten(\cdot)$ flattens a two-dimensional matrix into a one-dimensional vector, and $Conv_{1\times1}$ represents plain convolution operation with kernel size of 1×1.

To obtain patch-level information x_{pa}, we first aggregate all the pixels in a local patch, by applying a convolution operation with kernel size of $g \times g$, where $g \times g$ is equal to the size of initial superpixel. It is formulated as below,

$$x_{pa} = Conv_{g\times g}(x) \tag{7}$$

where $Conv_{g\times g}(\cdot)$ denotes convolution operation with kernel size of $g \times g$ and stride g. Then, the patch-level key vector K_{pa} and value vector V_{pa} can be obtained by applying flatten and linear transformation operation, which is formulated as follows,

$$\begin{aligned} K_{pa} &= Conv_{1\times1}(flatten(x_{pa})) \\ V_{pa} &= Conv_{1\times1}(flatten(x_{pa})) \end{aligned} \tag{8}$$

The Pixels-Patches Affinity Matrix $Affn_{Pi2Pa}$ can be obtained by analyzing the correlation between pixels and corresponding patches, which is formulated as below,

$$Affn_{pi2pa}^{L/G} = (Q_{pi}^T K_{pa}^{L/G})/\sqrt{(d)} \tag{9}$$

where L refers to the local attention that is computed between pixels and 9 nearest neighbor surrounding patches. The G refers to the global attention that is computed between pixels and all patches. d is the channel dimension. Then, the central pixel feature $\widetilde{x}^{L/G}$ can be updated as follows,

$$\widetilde{x}^{L/G} = V_{pa}^{L/G} softmax_{-1}(Affn_{pi2pa}^{L/G})^T \tag{10}$$

where $softmax_{-1}$ denotes the softmax operation in last dimension.

Local Attention Information Aggregation. As shown in top of Fig. 4(c), to aggregate the local patches information x^L, we conduct the local Pixels-Patches Self Attention, by analyzing the relationship between pixels and surrounding 9 nearest neighbor patches, which is formulated as follows,

$$\begin{aligned} \widetilde{x}^L &= PiPaSA^{9NNPatches}(x) \\ x^L &= Conv_{1\times1}(reshape(\widetilde{x}^L)) \end{aligned} \tag{11}$$

where $PiPaSA^{9NNPatches}(\cdot)$ denotes a concise local Pixels-Patches Self Attention representation of Eq.(6)-(10).

Global Attention Information Aggregation. As shown in bottom of Fig. 4(c), to aggregate the global patches information x^G, we conduct the global Pixels-Patches Self Attention, by analyzing the relationship between pixels and all patches, which is formulated as follows,

$$\begin{aligned} \widetilde{x}^G &= PiPaSA^{AllPatches}(x) \\ x^G &= Conv_{1\times1}(reshape(\widetilde{x}^G)) \end{aligned} \tag{12}$$

where $PiPaSA^{AllPatches}(\cdot)$ denotes a concise global Pixels-Patches Self Attention representation of Eq.(6)-(10).

Finally, the local context x^L, global context x^G and original features are fused to achieve superpixels-friendly features x^{LG}. The fusion computation can be formulated as follows,

$$x^{LG} = GELU(BN(Conv_{1\times1}(cat(x^L, x^G, x))) \tag{13}$$

where $cat(\cdot)$ refers to the concatenation operation in channel dimension.

D. Pixels to Superpixels module (P2SP)

To generate the final superpixels, we adopt a soft k-means algorithm similar to the ones in [20], [38] to determinate the affiliation of each pixel to its adjacent superpixels. As shown in Fig. 1, the P2SP module has two input features: $X_1 \in \mathbb{R}^{H\times W\times C}$ generated from decoder is used as the final superpixel-friendly features, and $X_5 \in \mathbb{R}^{h\times w\times c'}$ generated from stage5 is used to initialize the superpixel features. For the convenience of calculation, these features are first transformed by applying Eq.(14)-(15), obtaining $S_0 \in \mathbb{R}^{N_{patch}\times 9\times C}$ and $X_{pi} \in \mathbb{R}^{N_{patch}\times N_{pixel}\times C}$.

$$S_0 = unfold(Conv_{1\times1}(X^5)) \tag{14}$$

$$X_{pi} = DiRearrange(X^1) \tag{15}$$

where $N_{patch} = h \times w$ is the number of patches (initial superpixels). $Conv_{1\times1}(\cdot)$ is used to align the channel dimension. $unfold(\cdot)$ is used to generate 9 nearest neighbor patch features. $N_{pixel} = \frac{H}{h} \times \frac{W}{w}$ is the number of pixels within one patch. $DiRearrange(\cdot)$ refers to the dimension rearrange function. Then, the pixel assignment matrix $Assn \in \mathbb{R}^{N_{patch}\times N_{pixel}\times 9}$ is generated using Eq.(16),

$$Assn = softmax_{-1}(\frac{X_{pi}S_0^T}{\sqrt{d}}) \tag{16}$$

where $softmax_{-1}$ denotes the softmax in the last dimension, and d is the channel number. We further rearrange $Assn$ with shape $H \times W \times 9$, where the dimension 9 represents the probability of current pixel belonging to its 9 nearest neighbor superpixels. Finally, the pixel assignment can be obtained using Eq.(17),

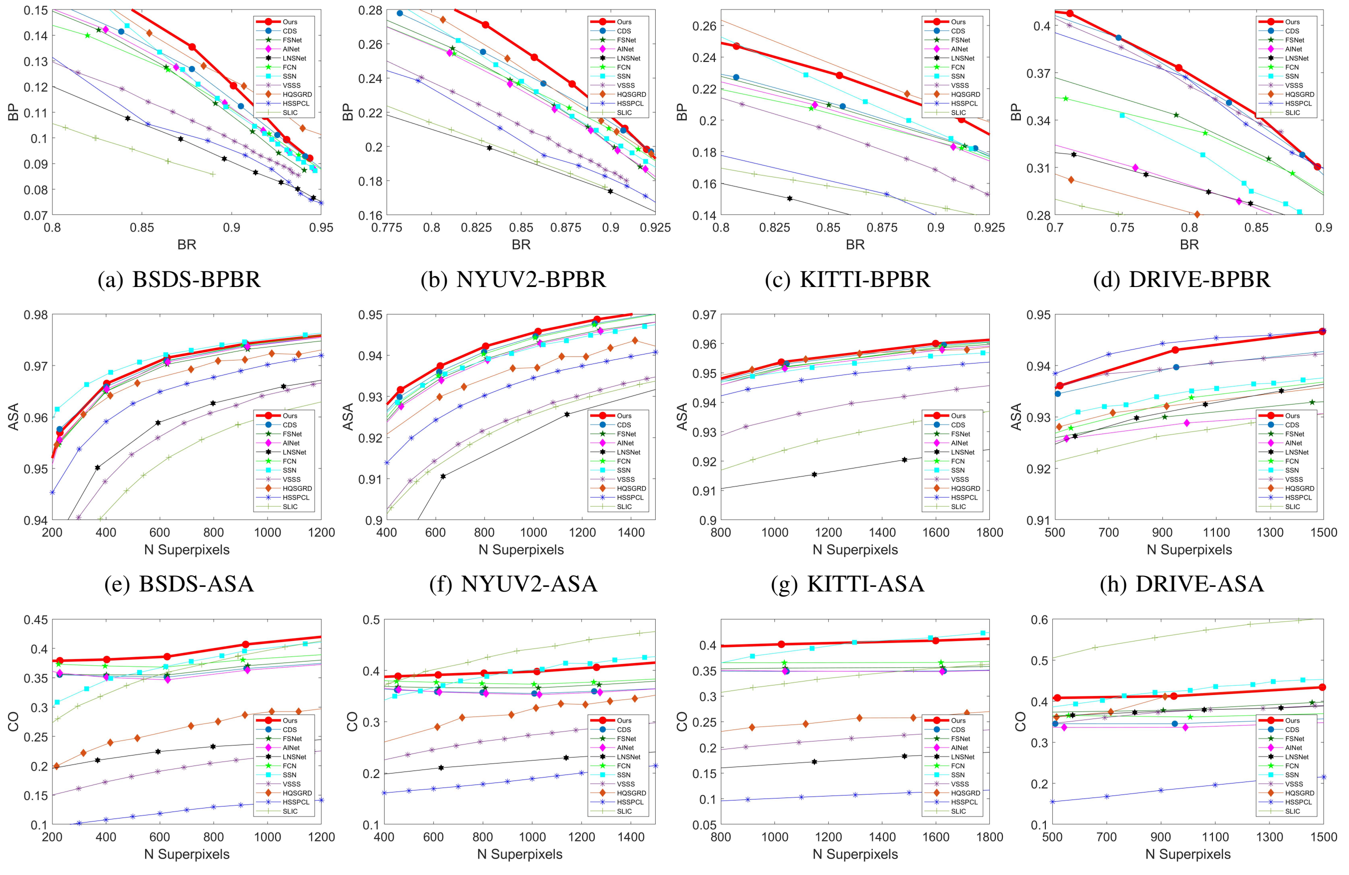
$$H_{i,j}^P = \underset{P\in\{1,2,\ldots,9\}}{argmax}\ (Assn(i,j), P) \tag{17}$$

where $argmax$ is the maximum value parameter function. $H_{i,j}^P$ represents the (i,j)th pixel belongs to the Pth patch.

E. Training Loss

The pixel soft assignment matrix $Assn \in R^{H\times W\times 9}$ plays a key role in the conversion between pixels and superpixels, which is expected to be optimized in training stage. Since there is no such groundtruth labels for the assignment matrix, we adopted a similar strategy in [20], [21] to indirectly learn $Assn$, by minimizing the distance between reconstructed pixel-wise property and groundtruth labels. Specifically, we first produce the superpixel-level property f_{sp} as follows,

$$f_{sp} = \frac{\sum_{p:sp\in\mathcal{N}_p} f_p \cdot Assn(p, sp)}{\sum_{p:sp\in\mathcal{N}_p} Assn(p, sp)} \tag{18}$$



(a) BSDS-BPBR (b) NYUV2-BPBR (c) KITTI-BPBR (d) DRIVE-BPBR

(e) BSDS-ASA (f) NYUV2-ASA (g) KITTI-ASA (h) DRIVE-ASA

(i) BSDS-CO (j) NYUV2-CO (k) KITTI-CO (l) DRIVE-CO

Fig. 5: Performance comparison with ten SOTA models on four datasets. From left to right: BSDS500, NYUV2, KITTI, DRIVE datasets. From top to bottom: BP-BR, ASA, CO score.

where $\mathcal{N}_p$ represents the neighboring superpixels of pixel p, f_p refers to the original pixel-wise property, and $Assn(p, sp)$ denotes the assignment matrix between pixels p and superpixels sp. Subsequently, we reconstruct pixel-wise property f'_p from superpixel-level property f_{sp} and $Assn(p, sp)$ as follows,

$$f'_p = \sum_{sp \in \mathcal{N}_p} f_{sp} \cdot Assn(p, sp) \tag{19}$$

The pixel-wise properties include the semantic labels f_p^{sem} and spatial position coordinates $f_p^{x,y}$, which are optimized by the cross-entropy loss $\mathcal{L}_{CE}$ and $\mathcal{L}_2$ reconstruction loss, respectively. As formulated in Eq.(20),

$$\mathcal{L}_{\text{final}} = \mathcal{L}_{CE}(f_p^{\prime sem}, f_p^{sem}) + \lambda \mathcal{L}_2(f_p^{\prime x,y}, f_p^{x,y}) \tag{20}$$

where semantic labels f_p^{sem} encode pixel categories with one-hot encoding. The ‘GT Labels’ in Fig.7 show visual example, where grayscale values represent distinct semantic categories. $\lambda = m/S$ is a balance weight, S is the superpixel sampling interval and m is a balance weight. We set λ to $0.003/16$ in our model through detailed experiments.

IV. Experiments

A. Experimental Setting

1) Datasets: We evaluate our method on four benchmark datasets: BSDS500 [39], NYUV2 [40], KITTI [41], and DRIVE [42]. The BSDS500 comprises 200 training images, 100 validation images, and 200 test images, each annotated with multiple semantic labels. To align with prior research [20]–[22], [30], we treat each label as an independent sample. This results in a total of 1087 training samples, 546 validation samples, and 1063 test samples. NYUV2 is an indoor segmentation dataset that includes object instance labels. From its total of 1449 images, a subset of 400 test samples has been selected [43], by removing the ones with unmarked regions along object edges. KITTI is an autonomous driving scenarios segmentation dataset. It comprises 200 high-resolution images paired with corresponding semantic labels. DRIVE is a medical dataset for retinal vessel segmentation, which contains 20 images with vessel position labels. To ensure a fair comparison, we adopt a consistent training and testing protocol existed in previous works [20]–[22], [30]. Specifically, we train our model solely on the BSDS500 training set and evaluate its performance on the other datasets.

2) Implementation Details: We implement our method with PyTorch on a single NVIDIA RTX3090 GPU. We empirically adopted GELU activation function in our method for its smoother than ReLU, enhancing convergence and performance. In training stage, we randomly crop the images to 208×208 as input, and train our model for 150k iterations using the Adam optimizer (parameters β_1 and β_2 set to 0.9

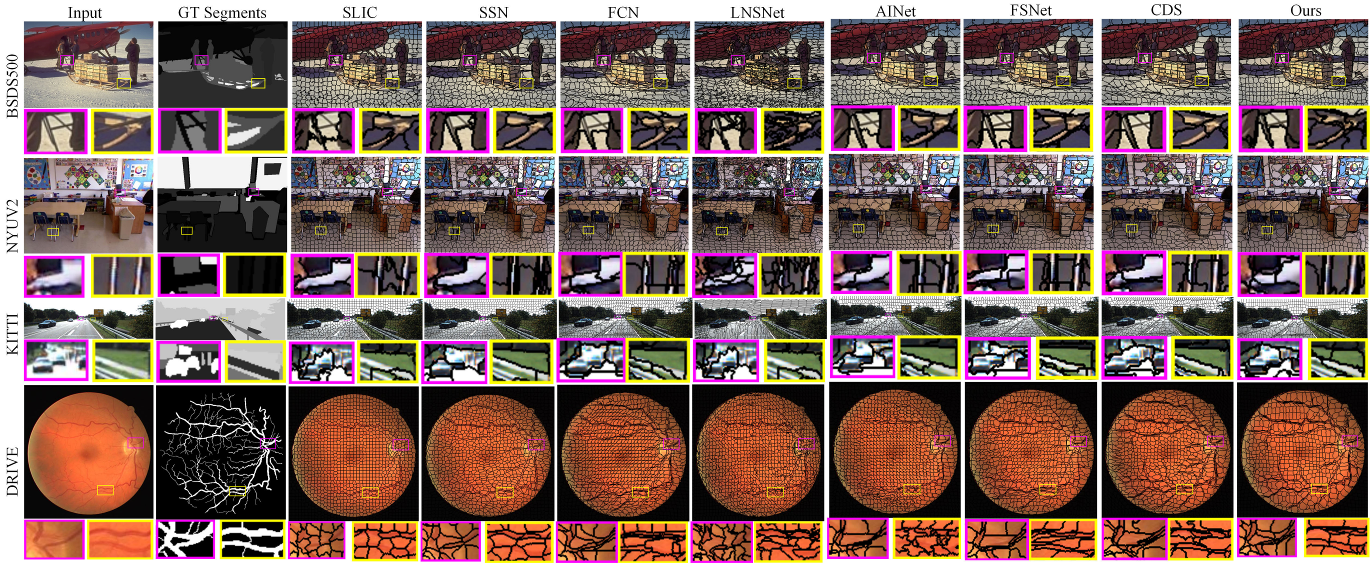


Fig. 6: Visual examples of superpixels generated by eight SOTA methods on BSDS500 dataset (top row), NYUV2 dataset (second row), KITTI dataset (third row), and DRIVE dataset (last row), respectively.

and 0.999). The batchsize is set to 16, and the initial learning rate is 8×10^{-4}, which is halved every 45k iterations. At training stage, the number of generated superpixels is 13×13, and the initial patches is with size of 16×16. After a forward inference, the initial superpixel feature of size 13×13 and the final pixel-level superpixel-friendly features of size 208×208 are utilized to compute the pixel-superpixel assignment matrix through a single association mapping operation. In testing phase, following previous works [21], different numbers of superpixels can be obtained by varying the input resolution.

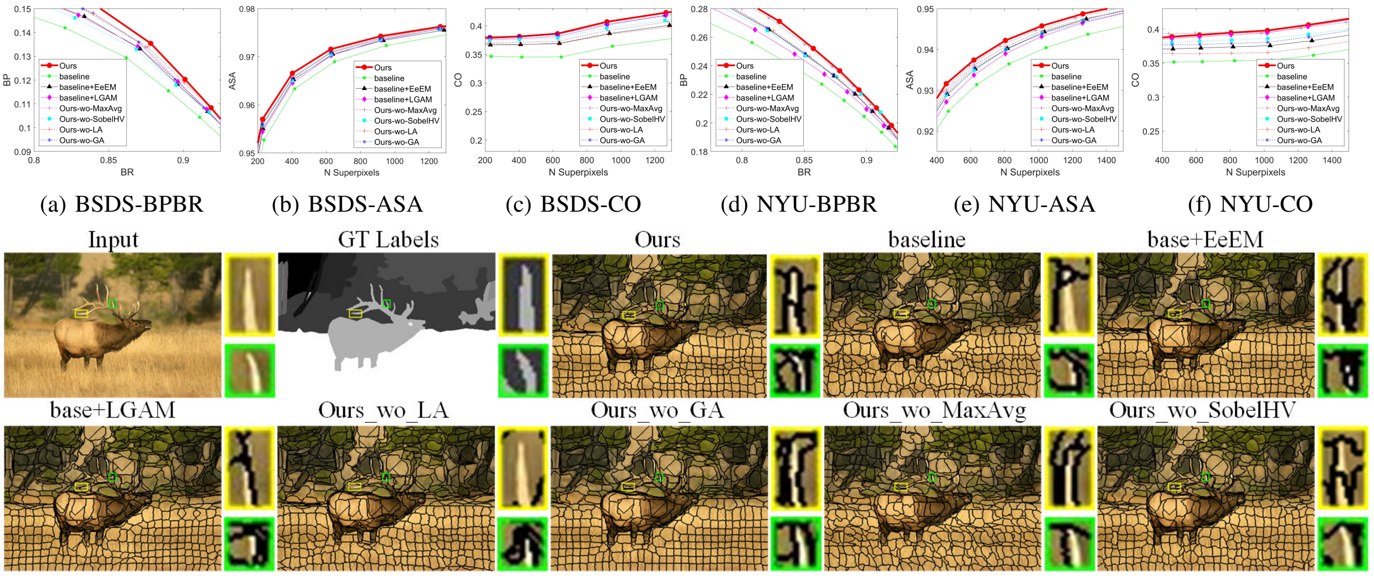
3) Evaluation Metrics: We assess our method using three popular metrics: Achievable Segmentation Accuracy (ASA), Boundary Precision and Boundary Recall (BP-BR), and Compactness Score (CO). ASA measures the upper limit of accuracy achievable by employing superpixels as a segmentation step. BP-BR assesses the correspondence between segmentation outcomes and ground truth boundaries. And the CO score indicates the compactness of the superpixels. All metrics obey that higher scores indicate better results.

B. Comparison with the state-of-the-art models.

In this part, we compare our method with 10 state-of-the-art models, including 4 traditional methods VSSS [29], HQSGRD [44], HSSPCL [23], SLIC [2], and 6 deep learning methods CDS [45], FSNet [33], LNSNet [46], AINet [22], FCN [21], SSN [20]. SLIC is the classic traditional method, and HSSPCL, HQSGRD, VSSS are 3 most recent advanced traditional methods. CDS, FSNet, LNSNet, AINet, FCN, and SSN are 6 representative deep learning methods. All results are either collected from publicly available sources or generated by executing the released source code.

1) Quantitative Comparisons: The quantitative results on four popular datasets are presented in Fig.5. From these results, it is evident that our method generally achieves the best or competing performance across all datasets, in terms of all three metrics. For the edge accuracy, our method generally achieves advanced performance on BSDS500, NYUV2, KITTI, and DRIVE datasets, in terms of ASA and BP-BR. Although the advanced HQSGRD, SSN and HSSPCL achieve a slightly higher performance than ours in Fig.5(c),(e),(h), our method outperforms them by a large margin on other datasets or metrics. These comparisons highlight the superiority of our method in preserving accurate edge information. As shown in Fig.5(i),(j),(k),(l), regarding the compactness (CO) of superpixels, SLIC achieves a slightly higher performance than ours on NYUV2. This phenomenon can be attributed to the fact that NYUV2's indoor scenes shown in second line of Fig.6, characterized by simpler textures and planar structures, favor SLIC's clustering algorithm, resulting in more compact superpixels. For the DRIVE dataset, in terms of compactness, SLIC achieves significantly higher performance than our method, but in terms of BPBR and ASA, the SLIC method lags far behind ours. This is because DRIVE is an eye vessel dataset that contains various relatively fine vessels, and SLIC cannot perceive these vessels well, mistakenly identifying them as uniform backgrounds, resulting in excessively high compactness at the cost of sacrificing much edge perception performance. This phenomenon can also be observed from Fig.6, the 3rd column represents the results of SLIC and 10th column represents the results of our ELGANet. While SLIC achieves higher compactness, it fails to effectively perceive edge information. Our method both retains edge information well and achieves good compactness. Despite this, our method achieves the best or competing performance across all datasets. This can be attributed to the effectiveness of our new local-global attention module, which facilitate the perception of similar pixels and contribute to compactness improvement. Additionally, our method is trained solely on the BSDS500 training set and evaluated on other datasets without fine-tuning. This further validates the strong generalization capability of our method.

2) Qualitative Comparisons: As illustrated in Fig.6, we gather some visual examples of seven representative models



(a) BSDS-BPBR (b) BSDS-ASA (c) BSDS-CO (d) NYU-BPBR (e) NYU-ASA (f) NYU-CO

(g) Visual examples generated by different network variants.

Fig. 7: Ablation studies on BSDS500 and NYUV2 dataset. (a)-(f) show the quantitative ablation results on both two datasets in terms of BPBR, ASA and CO metrics. (g) shows the visual examples generated by different network variants.

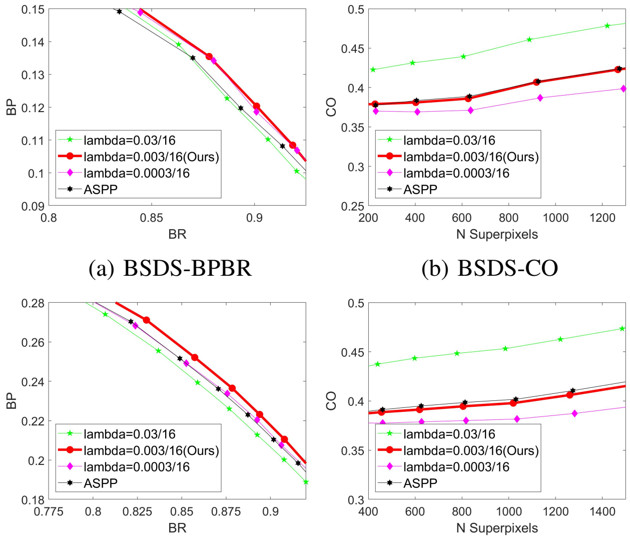
across BSDS500, NYUV2, KITTI, and DRIVE datasets. To maintain clarity amidst variations in image resolution across these datasets, we customize the number of generated superpixels as follows: 600 for BSDS, 800 for NYUV2, 1000 for KITTI, and 1600 for DRIVE. Comparing the visual examples of seven SOTA models, our method exhibits superior performance in preserving accurate boundaries and achieving a higher degree of superpixel compactness. In the visual example of DRIVE shown in Fig.6, when compared to FCN, LNSNet, AINet, FSNet, and CDS, our method (displayed in the last column) not only effectively preserves the vessel's edge but also maintains an impressive level of compactness. In contrast, although SLIC and SSN exhibit superior compactness, they compromise on the accuracy of perceiving the vessel's edge, leading to insufficient vessel perception. For the visual example of NYUV2 shown in Fig.6, our method can better retain the legs of the chair and achieve better compactness. Similarly, the visual examples of BSDS500 and KITTI can also draw the same conclusion. These observations collectively underscore the superiority of our ELGANet.

C. Model Analysis

1) Ablation Study: To validate the effectiveness of our new EeEM and LGAM, we conducted comprehensive ablation studies on the BSDS500 and NYUV2 datasets. Specifically, we first trained 4 network variants with different configurations using BSDS500 training set, and evaluated them on both BSDS500 test set and NYUV2. The network variants include: 'baseline' is the basic network without any additional modules; 'baseline+EeEM' is the basic network equipped with EeEM; 'baseline+LGAM' is the basic network equipped with LGAM; 'Ours' is our full model, which is equivalent to 'baseline+EeEM+LGAM'. The solid lines in Fig. 7(a)-(f) depict the quantitative results. 'baseline+EeEM' has achieved a higher BPBR, ASA and CO performance compared to 'baseline', which validates the effectiveness of our EeEM. Similarly, 'baseline+LGAM' also shows performance enhancement, particularly in significantly improving the CO score, compared to 'baseline'. These observations affirm the effectiveness of LGAM in enhancing superpixel compactness. Finally, the 'Ours' achieves the best performance on both metrics, showcasing the effectiveness of both EeEM and LGAM. The visual examples in Fig. 7(g) further illustrate the consistent performance improvement.

Additionally, we also trained another 4 network variants to validate the effectiveness of the key components within EeEM or LGAM. 'Ours_wo_MaxAvg' refers to the variant where we remove the 'MaxAvg' component from our EeEM; 'Ours_wo_SobelHV' refers to the variant where we remove the 'SobelH-DWConv' and 'SobelV-DWConv' branches from our EeEM; 'Ours_wo_LA' refers to the variant where we remove the local attention module from our LGAM; 'Ours_wo_GA' refers to the variant where we remove the global attention module from our LGAM. The dashed lines in Fig.7(a)-(f) depict the quantitative results. It can be observed that removing any component from our network results in a performance decrease, underscoring the pivotal role of our novel components. The visual examples in Fig.7(g) further support these consistent conclusions.

2) Comparison between LGAM and ASPP: To validate the effectiveness of our proposed LGAM in modeling contextual information, we replace the LGAM with the classical ASPP (Atrous spatial pyramid pooling) module, which comprises multiple parallel convolutions with varying dilation rates. As shown in Fig.8, the black line represents the model variant equipped with ASPP, while the red bold line denotes the model equipped with LGAM. As can be seen from Fig.8 (b),(d), our method achieves similar performance to ASPP in terms



(a) BSDS-BPBR (b) BSDS-CO

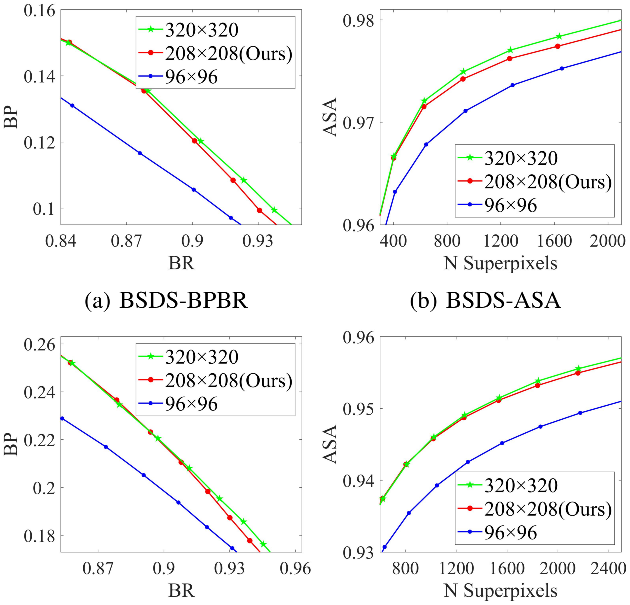
(c) NYUV2-BPBR (d) NYUV2-CO

Fig. 8: The comparison of different global feature extractors, and the impact of different λ values on performance.

of compactness (CO metric). However, as shown in Fig.8 (a),(c), our LGAM significantly outperforms ASPP in terms of BPBR metric. These observations support the superiority of our proposed LGAM in modeling contextual information.

3) The impact of hyperparameter λ in loss function: To examine the impact of the hyperparameter λ in the loss function Eq.(20) on performance, we analyzed the effects of varying balancing weights on performance. As shown in Fig.8 (b),(d), as the λ increases, the compactness value also increases. This can be attributed to the fact that a larger λ value favors the formation of superpixels among pixels that are closer in spatial position. As depicted in Fig.8 (a),(c), it can be observed that excessively large or small values of λ can both lead to a decline in performance. The optimal performance is achieved when λ is set to $0.003/16$. Therefore, considering the balance between compactness and accuracy, we set a λ value to $0.003/16$ in all experiments.

4) The impact of different image sizes on performance: We further investigated the impact of image size on ELGANet's performance. Three model variants are trained using three different image sizes (e.g., 96×96, 208×208, and 320×320), and the quantitative results are shown in Fig. 9. We observe that as the image size increases, the overall performance improves. When the image size increases from the smaller 96×96 to 208×208, there is a significant performance boost. However, when the image size increases from 208×208 to the larger 320×320, the performance improvement becomes more modest. This may be due to the fact that as the image size increases, it begins to provide richer spatial detail, but beyond a certain size, the additional details become limited, leading to slower performance gains. In addition, processing high-resolution images incurs significantly higher computational costs. To balance segmentation performance and computational efficiency, we set the default image size to 208×208 in our experiments.



(a) BSDS-BPBR (b) BSDS-ASA

(c) NYUV2-BPBR (d) NYUV2-ASA

Fig. 9: The impact of different image sizes on performance.

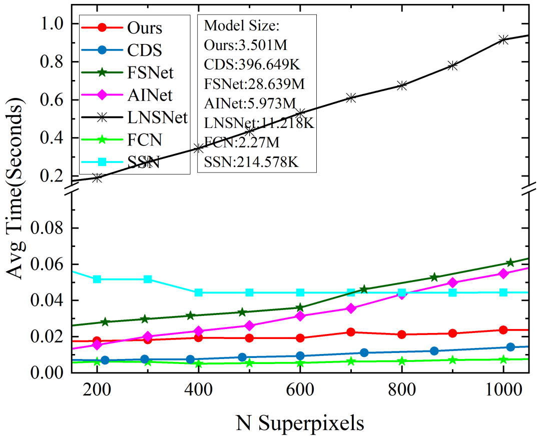


Fig. 10: Complexity comparison between different methods

D. Model Complexity

We also make a comparison of inference speed and model size between our method and six deep networks. For a fairness, we tested on BSDS500 using the same workstation equipped with a NVIDIA RTX3090 GPU. As shown in Fig.10, our inference speed is slightly lower than that of FCN and CDS methods, yet it is significantly superior to the other four methods. In addition, our model comprises 3.501M parameters, slightly higher than those of SSN, FCN, LNSNet, and CDS, but all these models remain within a relatively low parameter range. Despite owning a complexity comparable to or slightly higher than that of other models, our model generally outperforms them across all four datasets in terms of super-pixel segmentation performance, also demonstrating the superiority of our method.

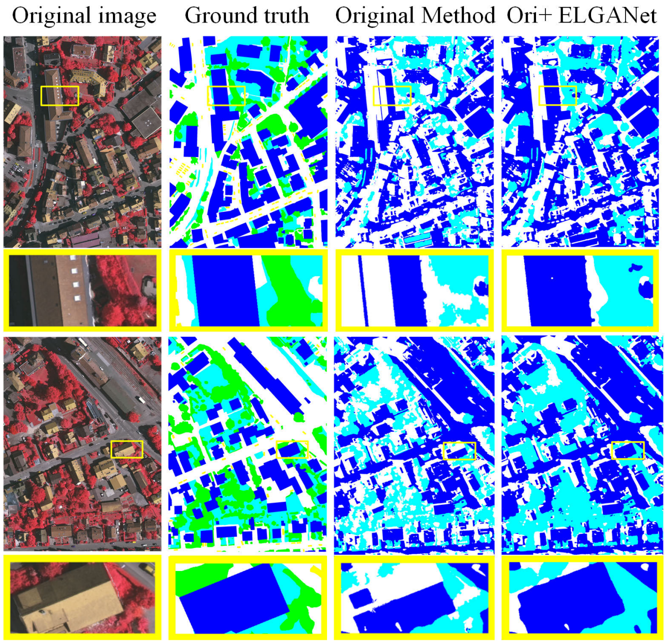


Fig. 11: Visual examples of the application in USSS.

TABLE I: Performance comparison between Unsupervised Semantic Segmentation and our ELGANet-assisted Unsupervised Semantic Segmentation. The higher the metric value, the better. The upward arrow indicates the performance improvement.

Metrics	Original Method	Ori+ELGANet	Improvement
Average AMI	0.240	0.316	0.076 (↑)
Average FMI	0.472	0.530	0.058 (↑)

E. Application

Superpixel can be applied in image semantic segmentation task. In unsupervised semantic segmentation (USSS) of remote sensing images [47], superpixels play a crucial role as pseudo-labels for unsupervised learning. To examine the advantages of our superpixel segmentation method in the semantic segmentation of remote sensing images, we replace the superpixels in the original method with our proposed ELGANet. The experimental results are shown in the Table.I. We evaluated the performance using two key metrics: Average AMI and Average FMI (Please refer to [47] for specific details). By integrating our ELGANet into the original method, we observed a significant improvement in these performance indicators. The visual examples of semantic segmentation are shown in Fig.11, clearly demonstrating that Ori+ELGANet achieves superior pixel-level semantic segmentation results compared to the original method. All these observations strongly validate the effectiveness and importance of utilizing superpixels to enhance image segmentation tasks.

F. Failure cases

Our method achieved general superior performance on four different datasets. However, our ELGANet still performs unsatisfactorily in complex scenarios with low contrast. Fig.12 exhibits several failure examples, where the foreground and background share similar color and texture characteristics. In these cases, our method clusters pixels belonging to different categories into the same superpixel, failing to preserve the edge information of the object well. This may be attributed to the blurred boundary characteristics, which prevent the clear distinction of different pixels. In the future, we plan to further investigate solutions for low-contrast failure scenarios, potentially incorporating advanced attention mechanisms or adaptive loss functions, which hold great potential for enhancing performance.

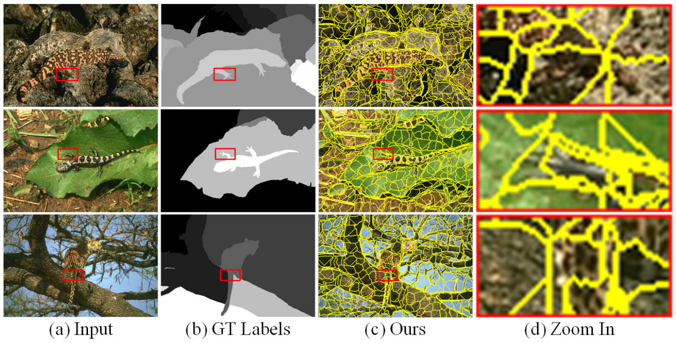

(a) Input (b) GT Labels (c) Ours (d) Zoom In

Fig. 12: Some visual examples of the failure cases.

V. Conclusion

To enhance the performance of superpixels segmentation, we propose a novel Edge guided Local-Global Attention Network (ELGANet), which incorporating with our novel Edge Enhancement Module (EeEM) and Local-Global Attention Module (LGAM). In EeEM, multiple types of edge information are leveraged to enhance the network's ability of preserving precise object edge, thereby improving the accuracy of superpixel segmentation. In LGAM, we explore local and global context interactions between pixels and patches to identify similar pixels and enhance the superpixels compactness. Comprehensive ablation studies validate the effectiveness of our specially designed modules in improving performance. Experiments conducted on four popular datasets further demonstrate the superiority of our proposed method compared to ten state-of-the-art models. Although our ELGANet demonstrates promising performance, it remains suboptimal in low-contrast scenarios, and the computational complexity still requires further optimization. Therefore, in future works, we plan to conduct an in-depth investigation into these issues, aiming for further performance improvements.

References

[1] Ren and Malik, "Learning a classification model for segmentation," in *Proceedings Ninth IEEE International Conference on Computer Vision*, 2003, pp. 10–17 vol.1.

[2] R. Achanta, A. Shaji, K. Smith, A. Lucchi, P. Fua, and S. Süsstrunk, "Slic superpixels compared to state-of-the-art superpixel methods," *IEEE transactions on pattern analysis and machine intelligence*, vol. 34, no. 11, pp. 2274–2282, 2012.

[3] M. Xu, B. Liu, P. Fu, J. Li, and Y. H. Hu, "Video saliency detection via graph clustering with motion energy and spatiotemporal objectness," *IEEE Transactions on Multimedia*, vol. 21, no. 11, pp. 2790–2805, 2019.

[4] Y. Cai, L. Dai, H. Wang, L. Chen, and Y. Li, “A novel saliency detection algorithm based on adversarial learning model,” *IEEE Transactions on Image Processing*, vol. 29, pp. 4489–4504, 2020.

[5] M. Xu, P. Fu, B. Liu, and J. Li, “Multi-stream attention-aware graph convolution network for video salient object detection,” *IEEE Transactions on Image Processing*, vol. 30, pp. 4183–4197, 2021.

[6] Q. Sun, M. Liu, S. Chen, F. Lu, and M. Xing, “Ship detection in sar images based on multilevel superpixel segmentation and fuzzy fusion,” *IEEE Transactions on Geoscience and Remote Sensing*, vol. 61, pp. 1–15, 2023.

[7] J. Chen, H. Zhang, M. Gong, and Z. Gao, “Collaborative compensative transformer network for salient object detection,” *Pattern Recognition*, vol. 154, p. 110600, 2024.

[8] F. Yang, H. Lu, and M.-H. Yang, “Robust superpixel tracking,” *IEEE Transactions on Image Processing*, vol. 23, no. 4, pp. 1639–1651, 2014.

[9] J. Zhan, H. Zhao, P. Zheng, H. Wu, and L. Wang, “Salient superpixel visual tracking with graph model and iterative segmentation,” *Cognitive Computation*, vol. 13, pp. 821–832, 2021.

[10] C. Ouyang, C. Biffi, C. Chen, T. Kart, H. Qiu, and D. Rueckert, “Self-supervision with superpixels: Training few-shot medical image segmentation without annotation,” in *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XXIX 16*. Springer, 2020, pp. 762–780.

[11] S. Yi, H. Ma, X. Wang, T. Hu, X. Li, and Y. Wang, “Weakly-supervised semantic segmentation with superpixel guided local and global consistency,” *Pattern Recognition*, vol. 124, p. 108504, 2022.

[12] T. C. Ng, S. K. Choy, S. Y. Lam, and K. W. Yu, “Fuzzy superpixel-based image segmentation,” *Pattern Recognition*, vol. 134, p. 109045, 2023.

[13] Z. Li and J. Chen, “Superpixel segmentation using linear spectral clustering,” in *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2015, pp. 1356–1363.

[14] Y.-J. Liu, C.-C. Yu, M.-J. Yu, and Y. He, “Manifold slic: A fast method to compute content-sensitive superpixels,” in *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016, pp. 651–659.

[15] P. F. Felzenszwalb and D. P. Huttenlocher, “Efficient graph-based image segmentation,” *International journal of computer vision*, vol. 59, pp. 167–181, 2004.

[16] L. Grady, “Random walks for image segmentation,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 28, no. 11, pp. 1768–1783, 2006.

[17] M.-Y. Liu, O. Tuzel, S. Ramalingam, and R. Chellappa, “Entropy rate superpixel segmentation,” in *CVPR 2011*, 2011, pp. 2097–2104.

[18] X. Kang, L. Zhu, and A. Ming, “Dynamic random walk for superpixel segmentation,” *IEEE Transactions on Image Processing*, vol. 29, pp. 3871–3884, 2020.

[19] J. An, Y. Shi, Y. Han, M. Sun, and Q. Tian, “Extract and merge: Superpixel segmentation with regional attributes,” in *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XXX 16*. Springer, 2020, pp. 155–170.

[20] V. Jampani, D. Sun, M.-Y. Liu, M.-H. Yang, and J. Kautz, “Superpixel sampling networks,” in *Proceedings of the European Conference on Computer Vision (ECCV)*, 2018, pp. 352–368.

[21] F. Yang, Q. Sun, H. Jin, and Z. Zhou, “Superpixel segmentation with fully convolutional networks,” in *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020, pp. 13 961–13 970.

[22] Y. Wang, Y. Wei, X. Qian, L. Zhu, and Y. Yang, “Ainet: Association implantation for superpixel segmentation,” in *2021 IEEE/CVF International Conference on Computer Vision (ICCV)*, 2021, pp. 7058–7067.

[23] T. Yan, X. Huang, and Q. Zhao, “Hierarchical superpixel segmentation by parallel crtrees labeling,” *IEEE Transactions on Image Processing*, vol. 31, pp. 4719–4732, 2022.

[24] J. Zhao, R. Bo, Q. Hou, M.-M. Cheng, and P. Rosin, “Flic: Fast linear iterative clustering with active search,” *Computational Visual Media*, vol. 4, pp. 333–348, 2018.

[25] R. Achanta and S. Süsstrunk, “Superpixels and polygons using simple non-iterative clustering,” in *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017, pp. 4895–4904.

[26] H. Li, Y. Jia, R. Cong, W. Wu, S. T. W. Kwong, and C. Chen, “Superpixel segmentation based on spatially constrained subspace clustering,” *IEEE Transactions on Industrial Informatics*, vol. 17, no. 11, pp. 7501–7512, 2021.

[27] W. Jing, T. Jin, and D. Xiang, “Edge-aware superpixel generation for sar imagery with one iteration merging,” *IEEE Geoscience and Remote Sensing Letters*, vol. 18, no. 9, pp. 1600–1604, 2021.

[28] L. Sun, D. Ma, X. Pan, and Y. Zhou, “Weak-boundary sensitive superpixel segmentation based on local adaptive distance,” *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 33, no. 5, pp. 2302–2316, 2023.

[29] P. Zhou, X. Kang, and A. Ming, “Vine spread for superpixel segmentation,” *IEEE Transactions on Image Processing*, vol. 32, pp. 878–891, 2023.

[30] W.-C. Tu, M.-Y. Liu, V. Jampani, D. Sun, S.-Y. Chien, M.-H. Yang, and J. Kautz, “Learning superpixels with segmentation-aware affinity loss,” in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2018.

[31] L. Zhu, Q. She, B. Zhang, Y. Lu, Z. Lu, D. Li, and J. Hu, “Learning the superpixel in a non-iterative and lifelong manner,” in *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2021, pp. 1225–1234.

[32] S. Xu, S. Wei, T. Ruan, and Y. Zhao, “Esnet: An efficient framework for superpixel segmentation,” *IEEE Transactions on Circuits and Systems for Video Technology*, pp. 1–1, 2023.

[33] H. Li, J. Liang, W. Li, and W. Wu, “Fsnet: Frequency domain guided superpixel segmentation network for complex scenes,” in *Proceedings of the 31st ACM International Conference on Multimedia*, 2023, pp. 4129–4137.

[34] W. Jing, T. Jin, and D. Xiang, “Fast superpixel-based clustering algorithm for sar image segmentation,” *IEEE Geoscience and Remote Sensing Letters*, vol. 19, pp. 1–5, 2022.

[35] J. Hu, L. Shen, S. Albanie, G. Sun, and E. Wu, “Squeeze-and-excitation networks,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 42, no. 8, pp. 2011–2023, 2020.

[36] M. Tan and Q. Le, “Efficientnetv2: Smaller models and faster training,” in *International conference on machine learning*. PMLR, 2021, pp. 10 096–10 106.

[37] A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai, T. Unterthiner, M. Dehghani, M. Minderer, G. Heigold, S. Gelly *et al.*, “An image is worth 16x16 words: Transformers for image recognition at scale,” *arXiv preprint arXiv:2010.11929*, 2020.

[38] H. Huang, X. Zhou, J. Cao, R. He, and T. Tan, “Vision transformer with super token sampling,” in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2023, pp. 22 690–22 699.

[39] P. Arbeláez, M. Maire, C. Fowlkes, and J. Malik, “Contour detection and hierarchical image segmentation,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 33, no. 5, pp. 898–916, 2011.

[40] N. Silberman, D. Hoiem, P. Kohli, and R. Fergus, “Indoor segmentation and support inference from rgbd images,” in *Computer Vision–ECCV 2012: 12th European Conference on Computer Vision, Florence, Italy, October 7-13, 2012, Proceedings, Part V 12*. Springer, 2012, pp. 746–760.

[41] H. Abu Alhaija, S. K. Mustikovela, L. Mescheder, A. Geiger, and C. Rother, “Augmented reality meets computer vision: Efficient data generation for urban driving scenes,” *International Journal of Computer Vision*, vol. 126, pp. 961–972, 2018.

[42] J. Staal, M. D. Abràmoff, M. Niemeijer, M. A. Viergever, and B. Van Ginneken, “Ridge-based vessel segmentation in color images of the retina,” *IEEE transactions on medical imaging*, vol. 23, no. 4, pp. 501–509, 2004.

[43] D. Stutz, A. Hermans, and B. Leibe, “Superpixels: An evaluation of the state-of-the-art,” *Computer Vision and Image Understanding*, vol. 166, pp. 1–27, 2018.

[44] Y. Xu, X. Gao, C. Zhang, J. Tan, and X. Li, “High quality superpixel generation through regional decomposition,” *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 33, no. 4, pp. 1802–1815, 2023.

[45] S. Xu, S. Wei, T. Ruan, and L. Liao, “Learning invariant inter-pixel correlations for superpixel generation,” *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 38, no. 6, pp. 6351–6359, Mar. 2024.

[46] L. Zhu, Q. She, B. Zhang, Y. Lu, Z. Lu, D. Li, and J. Hu, “Learning the superpixel in a non-iterative and lifelong manner,” in *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2021, pp. 1225–1234.

[47] G. Chen, C. He, T. Wang, K. Zhu, P. Liao, and X. Zhang, “A superpixel-guided unsupervised fast semantic segmentation method of remote sensing images,” *IEEE Geoscience and Remote Sensing Letters*, vol. 19, pp. 1–5, 2022.

Mingzhu Xu (Member, IEEE) received the B.S., M.Sc., and Ph.D. degrees from the Harbin Institute of Technology (HIT), Harbin, China, in 2013, 2015, and 2021, respectively. He is currently an Assistant Professor with the School of Software, Shandong University, Jinan, China. His research interests include computer vision, multimedia computing, and information retrieval. Dr. Xu is also an Invited Reviewer for prestigious journals, including IEEE TMM, IEEE TCSVT, IEEE TKDE, IEEE TITS, Information Science, ACM MM, NeurIPS, ICML.

Zhengyu Sun is currently pursuing the B.S. degree with the School of Software, Shandong University, Jinan, China. His research interests include computer vision, referring image segmentation.

Yijun Hu is currently pursuing the B.S. degree with the School of Software, Shandong University, Jinan, China. Her research interests include computer vision, referring image segmentation, and visual saliency analysis.

Haoyu Tang (Member, IEEE) received the B.S. and Ph.D. degrees from Xi'an Jiaotong University, China, in 2016 and 2021, respectively. He is currently an Assistant Professor with the School of Software, Shandong University. His research interests include machine learning and multimedia retrieval.

Yupeng Hu (Member, IEEE) obtained his Ph.D. degree in Software Engineering from Shandong University, Jinan, China, in 2018. He is currently an Associate Professor with the School of Software, Shandong University. His research interests include information retrieval, data mining. Various parts of his work have been published in famous journals and forums, such as IEEE Transactions on Image Processing, Science China Information Sciences, and ACM Multimedia. He has served as a PC member for ACM MM, ACL, AAAI and a reviewer for IEEE TKDE and TMM.

Xuemeng Song (Senior Member, IEEE) received the B.E. degree from the University of Science and Technology of China, Hefei, China, in 2012, and the Ph.D. degree from the School of Computing, National University of Singapore, Singapore, in 2016. She is currently an Associate Professor with Shandong University, Shandong, China. She has authored or coauthored several papers in top venues, such as ACM SIGIR, MM, and TOIS. Her research interests include information retrieval and social network analysis. She is an AE of IEEE TCSVT, IEEE TMM, and a reviewer for many top conferences and journals.

Liqiang Nie (Senior Member, IEEE), Fellow of AAIA and IAPR, is currently the dean with the School of Computer Science and Technology, Harbin Institute of Technology (Shenzhen campus). He received his B.Eng. and Ph.D. degree from Xi'an Jiaotong University and National University of Singapore (NUS), respectively. His research interests lie primarily in multimedia content analysis and information retrieval. Dr. Nie has co-/authored more than 100 CCF-A papers and 5 books, with 20k plus Google Scholar citations. He is an AE of IEEE TKDE, IEEE TMM, IEEE TCSVT, ACM ToMM, and Information Science. Meanwhile, he is the regular area chair or SPC of ACM MM, NeurIPS, IJCAI and AAAI. He is a member of ICME steering committee. He has received many awards over the past three years, like ACM MM and SIGIR best paper honorable mention in 2019, the AI 2000 most influential scholars 2020, SIGMM rising star in 2020, MIT TR35 China 2020, DAMO Academy Young Fellow in 2020, SIGIR best student paper in 2021, first price of the provincial science and technology progress award in 2021 (rank 1), and provincial youth science and technology award in 2022. Some of his research outputs have been integrated into the products of Alibaba, Kwai, and other listed companies.